# Phase I SBIR Final Report
## "A Texture Synthesis Code for Cold Backgrounds"

Security Classification: Unclassified
Contract: DACA33-98-C-0004
COTR: Dr. George G. Koenig / CRREL
Performance Period: January 1998 – July 1998

Submitted By:

Prime Contractor: ThermoAnalytics, Inc.
Principal Investigator: Alan C. Koivunen

94X Airport Road
PO Box 66
Calumet, MI 49913
(906) 482-9560

August 11, 1998

**REPORT DOCUMENTATION PAGE**

*Form Approved*
*OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
11 August 1998

3. REPORT TYPE AND DATES COVERED
Final: January 1998 – July 1998

4. TITLE AND SUBTITLE
A Texture Synthesis Code for Cold Backgrounds

5. FUNDING NUMBERS
Contract DACA33-98-C-0004

6. AUTHORS
Alan Koivunen, Alex Kostinski, Tim Schulz

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
ThermoAnalytics, Inc.
94X Airport Road
PO Box 66
Calumet, MI 49913-0066

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Cold Regions Research and Engineering Laboratory
72 Lyme Road (George G. Koenig)
Hanover, NH 03755-1290

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

It is not an easy task to develop an approach to synthetic image generation which is both physically based and yet universal, because these are somewhat contradictory requirements to the extent that the physics of scattering off snow for a particular scene depends on the specifics of the scene. Our goal in Phase I was to extract universal physical features of all such scenarios while confining attention to coherent microwave scattering. The work was intended to provide a foundation upon which more sophisticated physical models could be used to generate the physical parameters needed to drive texture simulation, as these models become available. In Phase II we propose significant new work which includes: 1) extension of the background models; 2) further development of the capability to model mmw emission, visible scattering and IR emission; 3) application of current texture generation procedures to the extended bands and background types; 4) enhancement of texture synthesis capabilities along with validation; and 5) the development of a user-friendly interface for editing inputs to the physical models, running the background and scattering/emission models and displaying the synthetic texture.

14. SUBJECT TERMS
SBIR, texture generation, physics-based, synthetic scene, background models, infrared, millimeter wave, cold environments, snow model

15. NUMBER OF PAGES
44

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
Unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

19. SECURITY CLASSIFICATION OF ABSTRACT
Unclassified

20. LIMITATION OF ABSTRACT
SAR

## TABLE OF CONTENTS

# 1 Introduction

It is not an easy task to develop an approach to synthetic image generation which is both physically based and yet universal, because these are somewhat contradictory requirements to the extent that the physics of scattering off snow for a particular scene depends on the specifics of the scene. Our goal here was to extract universal physical features of all such scenarios while confining attention to coherent microwave scattering.

Our reasoning is based on a synthesis of ideas from diverse disciplines of most recent literature. The central theme (universal yet physically based) comes from the use of the Central Limit Theorem of probability as it applies to waves. The theorem is Central because of the universality of the results. This theorem leads directly to the speckle model for imaging of homogeneous backgrounds (e.g. see Goodman, 1985). This model has been recently verified experimentally for the case of homogeneous radar scattering off snow at grazing angles, Ulaby et al., 1998. We then extend this model by incorporating general user-prescribed correlation functions in order to simulate inhomogeneous patchy terrain and anisotropy is allowed as well.

Furthermore, the inhomogeneous case is modeled by the so-called K-distribution (to be developed in detail in Phase II). These probability density functions (pdf's) are universal even in the inhomogeneous case because they result from an application of modified Central Limit Theorem argument (it is a probability mixture model with Gamma distributed local mean). The argument is based on the random walk model but with the variable number of steps (see Jakeman and Pusey, 1978 and Jakeman and Tough, 1988). In particular, this point of view makes it transparent that one will seldom observe Gaussian intensity distributions in microwave imagery. The physics of the argument behind the new limit theorem and K-distributions is in the binomial distribution of the number of steps in the random walk. Unlike the Poisson distribution, such steps (elementary scattering acts) occur not independently but in bunches i.e. the steps are correlated (e.g. see Kostinski and Jameson, 1997 for applications in the atmospheric sciences). The universal character of the result can be seen from the fact that it has been successfully applied to recent observation in microwave scattering by atmospheric turbulence and by ocean surfaces.

The work was intended to provide a foundation upon which more sophisticated physical models could be used to generate the physical parameters needed to drive texture simulation, as these models become available. For example, current snow modeling capabilities are restricted to one dimension and produce no predictions of surface roughness characteristics – necessary for surface scattering predictions. Empirical relations between snow age and average grain size may be used to estimate a roughness, but additional effects (wind, for example) may produce a roughness anisotropy that current physical models are unable to predict.

The following subsections describe the work completed in the Phase I study

## 2 Terrain Geometry

We assume that the user will have data specifying terrain elevation and initial snow cover properties – essentially, SNTHERM.89 input data, over the terrain to be modeled. A "class number" is to be assigned to each terrain elevation point. The class number is an index specifying a particular SNTHERM.89 input file.

## 2.1 Generation of Faceted Terrain

The elevation data file is used to generate a faceted terrain description displayable with the FRED (Faceted Region EDitor) software (Curran, 1996). To generate the faceted terrain, elevation values are taken to be the *center* elevation points of rectangular "regions" forming the terrain mosaic, points ♦ in Figure 1. The elevations of the interior tile *corners,* marked •, are estimated by carrying out a bilinear interpolation between the centerpoint elevations. The elevations at the edges and corners of the full terrain, ∘, were determined by linearly extrapolating from the nearest terrain interior points. Each rectangle is then divided into two triangles. The rectangle surface normal is obtained by averaging the surface normals associated with each of the triangles.

At this point two approaches may be taken: (1) grouping the columns, within classes, by top surface normal direction, or (2) full modeling of the snow columns at each elevation point.

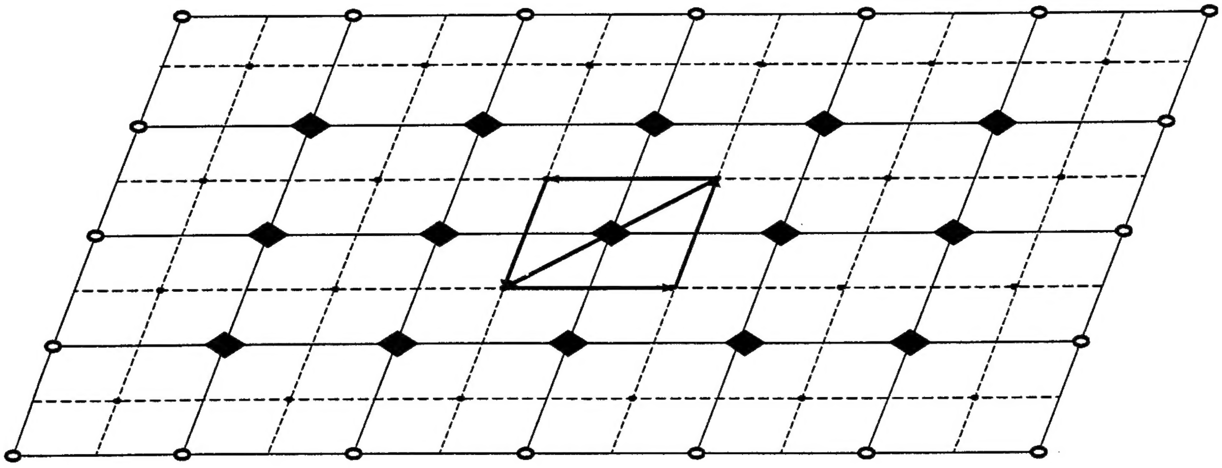


**Figure 1: Terrain elevation data at points marked ♦, extrapolated to points marked o and interpolated to points marked •.The quadrilateral in the center of the grid represents one snow column, the top surface of which is covered with two triangles forming one terrain tile. A surface normal vector is estimated for the top of this snow column by averaging the two surface normals associated with the triangles. This surface normal is used by *sntherm.***

## 2.2 Grouping of Similar Snow Columns to Reduce Computation

Simulation run time reduction may be accomplished by running SNTHERM.89 only on snow columns with dissimilar initial conditions and top surface slopes. If the terrain is very flat and the initial conditions are the same for each column making up the terrain, SNTHERM.89 need only be run once. The degree of grouping may be controlled by the user – using larger surface normal zenith and azimuth angle grouping intervals results in more parts of the terrain being represented by a single SNTHERM.89 run, and a coarser approximation to the terrain's actual evolution. For example, if the user specifies 10 degree azimuth and zenith angle intervals, all the snow columns with top surface normal vectors lying between 20 and 30 degrees zenith and 30 and 40 degrees azimuth would be represented by one SNTHERM.89 model. This model would have zenith angle of 25 degrees and azimuth angle 35 degrees. If the user used a 5 degree interval, the group would be split into four sub-groups, with the SNTHERM.89 zenith and azimuth angle closer to each of the snow columns actual azimuth and zenith angles – and consequent greater accuracy and run time. Software developed in Phase I examines the surface normals for all snow columns in a given initial condition class. If a snow column's surface normal lies within a particular zenith/azimuth interval, the snow column is then assigned to the appropriate group. Figure 2 displays an example faceted terrain surface generated from an artificial elevation file. Snow column 194 is highlighted.

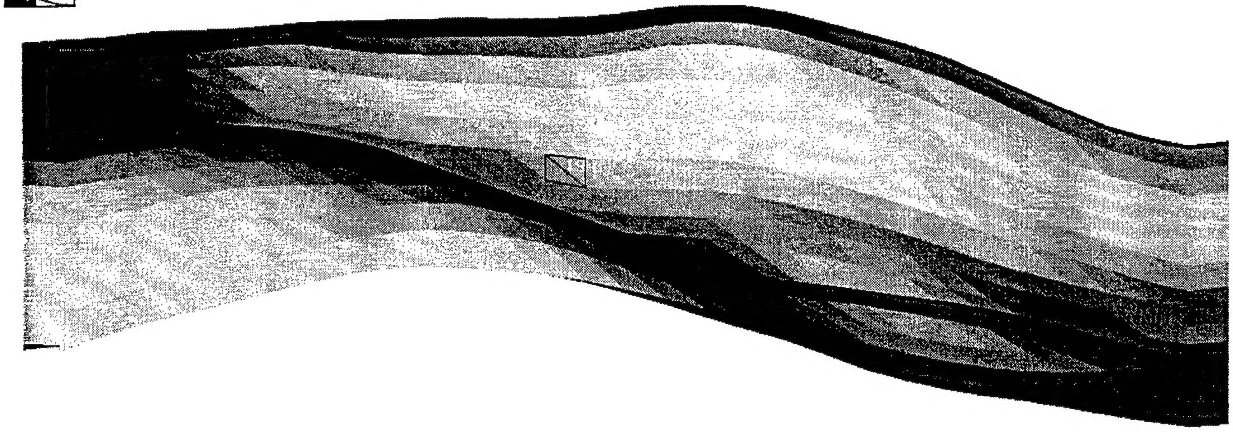


**Figure 2: Example faceted terrain generated from sample elevation file**

## 2.3 Modeling of Terrain Shading Effects

In this approach the faceted description of the terrain may be used with the FRED software (Curran et al. (1996)) to estimate the area of each terrain facet visible from each point over the sky dome. This "apparent area" calculation includes the effect of reduced area due to *cos(θ)* projection as well as shading of one part of the terrain by another. This shading effect is then included in a modified version of SNTHERM.89's subroutine *slope* to reduce the direct solar loading on the shaded snow column. Since the effect of shading on each snow column top surface changes as the sun travels across the sky, two columns with identical initial conditions and surface normal vectors may have dissimilar exposure to direct insolation at some point through the simulation. These two snow columns will evolve differently, and shouldn't be assigned to a single group. Grouping is still possible by pre-examining each snow columns' top surface solar exposure throughout the full simulation time and assigning those columns with similar exposure throughout the simulation to a single group. Terrains with low-relief will benefit most from grouping.

Modifications to SNTHERM.89 to model terrain shading effects have been included in the source code supplied with this final report, but have been disabled.

The reader may examine subroutine *slope.f* in the *TAI_Texture/Sntherm* directory, where the variable *shadowfactor* is used to scale direct solar radiation on the snow surface.

# 3 Snow Modeling

The most highly developed snow model of which we are aware is the one-dimensional temperature model for a snow cover (SNTHERM.89) developed at the U.S. Army Cold Regions Research and Engineering Laboratory. With slight modifications it provided the predictions of time varying snow characteristics (layer thickness, average grain size, water content and mass density by layer) used to control the mmw backscatter coefficient calculations.

## 3.1 A Description of SNTHERM.89

SNTHERM.89 is a research grade one-dimensional mass and energy balance model for predicting temperature profiles within strata of snow and frozen soil. It addresses conditions ranging from initial ground freezing in the fall to snow ablation in the spring. It is adaptable to a full range of meteorological conditions such as snowfall, rainfall, freeze-thaw cycles and transitions between bare and snow-covered ground. Snow cover densification and metamorphosis and their resulting impact on optical and thermal properties are included, as well as the automatic treatment of snow accumulation and ablation. Transport of liquid water and water vapor are included as required components of the heat balance equation and modeled assuming gravitational flow. It extends to the saturated case of water ponding on ice lenses or frozen soil. Phase-change, water flow and temperature are coupled through the use of a freezing curve. Although the model is primarily intended for use in snow, it will accommodate the bare soil case. The fluid-flow algorithm, however, does not include the effects of capillary pressure and consequently does not model water flow in soil.

The model handles five different material types or layers. A numerical solution is obtained by subdividing snow (and soil) layers into horizontally infinite control volumes. Each of which is then subject to the governing equations for heat and mass balance. The control volume approach of Patankar (1980) is used to divide the full volume into spatially discrete control volumes. Governing sets of equations are linearized with respect to unknown variables and then solved. An adaptive time-step is used that automatically adjusts between maximum and minimum values to achieve the desired solution accuracy. The user inputs these maximum and minimum time-step values. Mesh thickness is constrained to a minimum of 2 mm, and to a maximum of 1.67 and 3.33 cm for the top two nodes.

The governing equations are subject to meteorologically determined boundary conditions at the air-snow interface. Surface fluxes are computed from user-supplied meteorological observations of air temperature, dew point, wind speed and precipitation. If available, measured values of solar and infrared radiation are used. In the absence of radiation data, the model provides estimations using routines that take into account location, solar aspect, cloud conditions and the albedo and inclination of the surface. Furthermore, any of the meteorological values can be determined by user supplied functions. The model is initialized with profiles of temperature and water content for the various strata, the accuracy of which determines the time required for the simulation to equilibrate after inception of the computer run. The user either enters physical characteristics for the selected strata or optionally supplied from internal databases provided for snow (or sand and clay).

## 3.2 SNTHERM.89, modified

The texture synthesis procedure requires SNTHERM.89 snow property predictions over the entire terrain. With terrain divided up into snow columns via the method of section 2, and each terrain tile assigned an SNTHERM.89 class, we run SNTHERM.89 on each snow column. SNTHERM.89 has been embedded in a loop, running sequentially on each of the columns. The SNTHERM.89 source code was modified to re-initialize the appropriate variables at the start of each run.

SNTHERM.89 was not modified to model the full terrain at each time step, but rather must run through the full modeling scenario at each snow column before it proceeds to the next column. Since the backscatter coefficient calculations need snow property predictions for all snow columns at each time step, the SNTHERM.89 calculations need to be carried out completely – all snow columns at all times - before output is written. For this reason the interruption of *sntherm* before it is completed results in insufficient information for further texture synthesis calculations.

*SNTHERM.89* will be modified in Phase II to compute snow properties for all snow columns as the simulation passes through each time-step. This will permit texture synthesis up to the time at which *sntherm* is interrupted and allow intermediate results to be saved out. This will also allow a re-start of *sntherm* at the point of interruption.

Output subroutines have been modified to save information needed for subsequent modeling in the process. Snow properties for the top layer of each snow column are saved to **.rad* files for display with MuSES software. The predicted properties saved in these files include average snow grain size, snow mass density, and liquid water density. Since the backscatter coefficient calculations based on Narayanan and McIntosh, (1990) require snow properties for all layers, additional files are generated saving this information. Subroutine *writeCRREL* has been created to save out the additionally needed information.

# 4 Computation of Backscatter Coefficients

In order to simulate texture for snow backgrounds, we must make estimates of the backscatter coefficients for the terrain. We assume an active monostatic system. Since the signal we are considering has a wavelength of about 3mm and snow grains range from less than 0.5 mm to over 5mm in diameter (Narayanan and McIntosh, 1990), and the fractional volume of ice particles in a snow pack is appreciable, we should consider the effects of correlation in the scattered field.

Radar returns from background environments can be attributed to two mechanisms: surface scattering and volume scattering. Surface scattering is dominant for returns from highly reflective terrains such as roads, ocean bodies and lakes. For general terrain features that are partially penetrable by electromagnetic waves – forests, vegetation canopies and snow and ice fields, volume scattering or the returns from distributed scatterers inside the medium, may contribute most. The relative importance of surface vs. volume scattering will also depend on the angle of incidence of the traveling wave. Normal incidence on a low density snow surface will result in significant volume scattering while scattering at grazing incidence, particularly if the snow is wet, will be dominated by scattering from the surface.

To compute the volume scattering contribution, we must begin with either a layered continuous random medium or a randomly distributed discrete scatterer characterization of terrain features. The distinction between the two characterizations lies in that the random medium model treats the random fluctuation of the permittivity tensor instead of dealing with individual scatterers.

The scattering problem can be formulated using full wave theory (Nghiem et al. 1990) or radiative transfer theory (Chandrasekhar, 1960).

Wave theory can include multiple scattering effects and is fully coherent, but in practice is limited by complexity in the formulation. It can currently be applied to a three-layer random medium model – with one of the regions defined by anisotropically random fluctuations of permittivity and one with isotropic randomness. The recent development of the distorted Born approximation (Tsang et al., 1985) technique and its application to this configuration greatly extends the range of validity of its solutions. This approximation takes into account the dissipation and scattering losses and the modification of wave speed due to embedded scatterers. Therefore multiple scattering has been considered to some extent. Physically, the first-order distorted Born approximation describes the single scattering process of the mean field.

The radiative transfer theory is constructed from the energy transport equation and deals directly with the wave intensities by assuming incoherence of the far field interactions. The effect of rough interfaces between layers can be taken into account through the boundary conditions. Concerned only with the intensities of electromagnetic waves, the radiative transfer theory ignores phase information, which is important to polarimetric remote sensing with monostatic radar. The problem of multilayered media has been considered for a two-layer configuration with Mie and Rayleigh scattering and for a three-layer configuration with a continuous random medium. The T-matrix method (Waterman and Truell, 1961) is used to compute the scattering function for individual non-spherical scatterers and a rotation matrix is used to relate the T-matrix to the principle frame for a given probability density of the Euler angles.

The composite rough surface model has been used in the past to calculate surface scattering returns. In this model, variation in the terrain profile is composed of different scales of roughness. The roughness can be randomly distributed or periodically varying. Different approximation techniques, including the Kirchoff Approximation (KA), the Small Perturbation Method (SPM), the extended boundary condition technique and Monte Carlo simulation techniques have been used with varying degrees of success for predicting surface scattering returns. The original Kirchoff approximation uses the tangent plane approximation for local surface fields and the physical optics (PO) integral. Its validity is therefore limited to relatively smooth surfaces.

Although a full-wave analysis may accurately account for this correlation, it is prohibitively complex. We instead consider the following two alternative approaches to backscatter coefficient calculation: (1) a combination of the Dense Medium Radiative Transfer (DMRT) and Rough Surface models used in the EMSARS radar simulation program and (2) a much simpler model based on geometric optics and Mie scattering theory.

## 4.1 EMSARS Based Volume Scattering

The radiative transfer theory, which is based on energy conservation, describes the attenuation and scattering of the random medium with extinction and phase matrices, respectively. When the scatterers inside the medium are sparsely distributed the extinction and phase matrices can be calculated by averaging the independent attenuation and scattering from individual scatterers, i.e. by neglecting correlation effects.

The EMSARS DMRT model treats dry snow as a mixture of air and ice particles. The wet snow model adds water particles to the medium. In a dense medium the fractional volume of more than one of the constituents is appreciable and the assumption of *independent scattering* often used in conventional radiative transfer theory is not valid. The effects of correlated scattering and the spatial correlation of the scatterers must be considered. To include the effects of mutual coherent interactions among closely spaced scatterers, the DMRT uses the quasi-crystalline approximation with coherent potential and the ladder approximation for correlated scatterers (Tsang, 1985).

In this model, snow is modeled as spherical ice particles with radius *a* and permittivity $\varepsilon_s = \varepsilon_s^{'} + \varepsilon_s^{''}$ embedded in a background medium, Liquid water, when present, is included as part of the background medium. We assume that the water particles are randomly oriented spheroids uniformly distributed in the snow. Thus the wet snow is modeled as a medium of ice particles in a "wet air" background. To determine the dielectric constant of this "wet air" background the Polder-van Santen mixing formula is used. Approximations for the effective propagation constant, extinction rate and albedo are derived and used in a modified form of the conventional radiative transfer equation. This equation is then solved using methods similar to that for conventional radiative transfer.

The DMRT model requires:

1. Permittivity of ice particles, which is quite stable at $3.15 + i0.0085$ for 95 GHz.

2. Permittivity of the underlying half-space – for frozen soil this is typically (6.0,0.6) $\varepsilon_0$.

3. Salinity and temperature of liquid water – the temperature is about 0 C and salinity about 0.0 to 0.01%.

4. Radius of the ice particle – this information is provided by *sntherm*.

5. Fractional volume of ice particle – also provided by *sntherm*

6. The snow wetness – from *sntherm*.

7. Thickness of the snow layer – *sntherm* estimates this.

The EMSARS DMRT model requires the specification of the size of water particles in the medium. We use the following relation between ice particle diameter and water particle diameter taken from Narayanan and McIntosh (1990):

$$\mathrm{d}_{\text{water}} = \mathrm{d}_{\text{ice}}\left(\frac{2m_v}{1-\phi-m_v}\right) \tag{1}$$

where $m_v$ is the fractional volume of liquid water, and $\phi$ is the porosity of the snow medium.

The model assumes scattering from a volume described by one set of the above properties – i.e. it doesn't model the effect of layering. We use the top layer properties predicted by *sntherm* and assume that these properties are constant through the full snow depth

The DMRT model is limited by its use of the Rayleigh phase matrix. Since the wavelength is on the same order as the particle diameter, Rayleigh scattering is a limiting approximation. The DMRT theory also assumes that the air-snow surface is flat and therefore makes no estimate of surface scattering, which is important at grazing angle incidence.

## 4.2 EMSARS Based Surface Scattering

The roughness of the interface between two different media affects the propagation and scattering characteristics of an incident wave. Numerous methods have been developed in the study of wave scattering from random rough surfaces. However, most analytic approaches for rough surface scattering are founded on certain approximations and are thereby limited in their range of application. Two classical theories in solving the rough surface scattering problem are the Small Perturbation Method (SPM) and the Kirchoff approximation (KA). In the small perturbation method, it is assumed that the rms height of a rough surface is much smaller than the incident wavelength. In the Kirchoff approximation, an approximate boundary condition is made on the unknown surface field with the physical optics approximation. The scattered field is then calculated from an exact diffraction integral.

Since surface scattering is so important, particularly at high angles of incidence and for rough, wet-snow surfaces, we also extracted the EMSARS rough surface model for use in estimating average backscatter coefficients for the terrain surface. EMSARS permits modeling rough surface scattering on two scales using a combination of the Kirchoff approximation for large scale roughness, where the radius of curvature of the roughness is assumed much longer than the wavelength, and the Small Perturbation Method (SPM) for small scale roughness. Since the SPM method assumes that the rms roughness is smaller than the incident wavelength, its application to snow covered surfaces may limit the validity of the model since rms roughness may be on the order of the grain diameter – and therefore about the same size as the wavelength.

The two–scale rough surface model is limited in that only single scattering from the surface is considered and the correlation function of the roughness is assumed to be Gaussian. Furthermore, its accuracy is limited in predicting scattering for grazing angle incidence.

Since *sntherm* provides no roughness estimates, we currently neglect large-scale roughness, setting the large-scale rms roughness to zero, and set the small-scale roughness to the average grain size. The small perturbation method also assumes that the slopes of the rough surface are relatively small – more specifically, the ratio (rms roughness height)/(correlation length) <<1. The total field is expressed as the sum of upward and downward traveling waves and the boundary conditions and the divergence relations determine the field amplitudes.

The total backscattered power is proportional to the sum of the surface and volume backscatter coefficients.

### 4.3 Narayanan and McIntosh based Scattering Estimates

The *scatter* model treats surface and volume scattering from layered snow configurations. The snow layers are assumed to consist of closely packed spherical scatterers of diameters ranging from some $d_{\min}$ to some $d_{\max}$. We assume, for simplicity that the diameters are uniformly distributed through this range and that $d_{\min}$ and $d_{\max}$ are simply related to the average snow particle diameter predicted by *sntherm* assuming a Rayleigh distribution of diameters. Other distributions will be implemented in Phase II. Surface scattering at the air-snow interface and at snow layer interfaces and volume scattering within each snow layer are all treated. The scatterers consist of ice particles in the case of dry snow and ice particles as well as water globules in the case of wet snow. The air-snow and snow-layer-snow-layer interfaces are assumed to have a known rms surface roughness $\sigma_{rs}$ and correlation length $\rho_l$. For simplicity we simply let $\sigma_{rs}$ and $\rho_l$ be a multiple of the *larger* average grain size ( *sntherm* predicted) of the two snow layers at the interface. It is also assumed that the electromagnetic signal suffers losses due to bistatic scattering at the interfaces and due to absorption within each snow layer volume. The total scattered power is represented by the incoherent sum of all the surface and volume scatter components, where it is assumed that each scattering mechanism is independent of the other and interactions between surface and volume scattering are neglected. That is, this is not a dense medium model.

The model treats multilayered snow in terms of a cascade of scattering from a number of homogeneous layers with distinct boundaries. The model is based on geometrical optics and Mie scattering theory with calculations of the permittivities of ice and water based on the work of Cole and Cole (1941). The permittivity of snow is estimated with a simple volume weighted formula using the permittivities of the constituents (air, ice and water) and information regarding the volume fractions of these constituents obtained from *sntherm* output.

The surface scatter at each dielectric interface is computed using geometric optics methods. It assumes a Gaussian distribution of slopes and a Gaussian profile correlation function. The surface backscatter coefficient between layers *i* and *j*, for co-polarized fields is then (Narayanan and McIntosh 1990)

$$\sigma_S^0 = \frac{|R(0)|^2}{4\left(\frac{\sigma_{rs}}{\rho_l}\right)^2 \cos^2(\Theta)} \exp\left[-\frac{\tan^2(\Theta)}{4\left(\frac{\sigma_{rs}}{\rho_l}\right)^2}\right] \tag{2}$$

where $\Theta$ is the angle of incidence, $R(0)$ is the Fresnel reflection coefficient for normal incidence, $\sigma_{rs}$ is the rms surface roughness between layers *i* and *j*, $\rho_l$ is the correlation length of the roughness between layers *i* and *j*. Again, since *sntherm* does not estimate rms roughness or correlation length, we let the user specify the correlation length and the inter-layer surface roughness, or set this roughness equal to the average grain size.

To compute the volume backscatter coefficient within each snow layer, we assume that the layer consists of spherical scatterers with Rayleigh size distribution, with an average value estimated by *sntherm.* Mie theory is then used to estimate the scattering cross section of each particle. We used the public domain BHMIE code (see Bohren and Huffman, 1998) to calculate the Mie cross-sections for both the ice and water particles. The overall backscatter coefficient for co-polarized scatter for dry snow is

$$\sigma_{\mathrm{V}}^{0} = \int_{d_{\min}}^{d_{\max}} n(D)\sigma_{\mathrm{M}}^{0}(D)\frac{\cos(\Theta)}{2\alpha_{\mathrm{E}}}[1-\exp(-2\alpha_{\mathrm{E}}h/\cos(\Theta))]\mathrm{d}D \tag{3}$$

Where *n(D)* is the number distribution of particles of diameter *D*, $\sigma_{\mathrm{M}}^{0}(D)$ is the Mie backscatter coefficient for a particle of diameter *D*, $d_{\min}$ and $d_{\max}$ are the minimum and maximum particle diameters over which to integrate, $h$ is the layer thickness and $\alpha_{\mathrm{E}}$ is the total attenuation coefficient per unit length through the snow volume. The corresponding coefficient for wet snow is

$$\sigma_{\mathrm{V}}^{0} = \frac{\cos(\Theta)}{2\alpha_{\mathrm{E}}}[1-\exp(-2\alpha_{\mathrm{E}}h/\cos(\Theta))]\cdot\left[\int_{d_{\min}}^{d_{\max}} n_i(D)\sigma_{\mathrm{Mi}}^{0}(D)\mathrm{d}D + \int_{d'_{\min}}^{d'_{\max}} n_w(D)\sigma_{\mathrm{Mw}}^{0}(D)\mathrm{d}D\right] \tag{4}$$

where the *i* and *w* subscripts indicate ice and water related parameters. The transmission coefficient at the interface of the *j'th* and *(j+1)'th* layers is given by

$$T = \left[1-|R(0)|^2\right]\cdot\exp\left(-\bar{\mathrm{q}}_{\mathrm{z}}^{2}\sigma_{rs}^{2}\right) \tag{5}$$

Where

$$\bar{q}_{z,j,j+1} = \frac{2\pi}{\lambda}\left\{\sqrt{\varepsilon'_{sj}} - \sqrt{\varepsilon'_{s,j+1}}\right\} \tag{6}$$

And $\varepsilon'_{sj}$ is the real part of the complex permittivity for snow in the *j'th* layer. The total backscatter coefficient at the surface of an N-layered snow column is then

$$\sigma^0 = \sigma^0_{sN,N+1} + T^2_{N,N+1}\left[\sigma^0_{VN} + L^2_N\left[\sigma^0_{S,N-1,N} + T^2_{N-1,N}\left[\sigma^0_{VN-1} + L^2_{N-1}\left[\ldots + L^2_1\sigma^0_G\right]\right]\right]\ldots\right] \tag{7}$$

Here $\sigma^0_{Sj,j+1}$ is the surface backscatter coefficient at the interface of the *j'th* and *(j+1)'th* layer, $\sigma^0_{Vj}$ is the bulk volume scatter coefficient within the *j'th* layer, $T_{j,j+1}$ is the power transmission coefficient at the interface of the *j'th* and *(j+1)'th* layers, $L_j$ is the power absorption coefficient within the *j'th* layer and $\sigma^0_G$ is the backscatter coefficient of bare ground.

The overall backscatter coefficient of equation (7) is modified to account for the shadowing effects of rough surfaces. Shadowing corrections are especially important at near grazing angles, where a large portion of the projected footprint falls within the shadow of the illuminated region. We write $\sigma^0_{total} = \sigma^0 S$ where (Narayanan and McIntosh (1990) )

$$S = \frac{1}{\left[\frac{2}{\sqrt{\pi}}\tan(\theta)\frac{\sigma_{rsj,j+1}}{\rho_{l,j,j+1}}\exp\left(-\frac{\rho^2_{1,j,j+1}}{4\sigma^2_{rsj,j+1}\tan^2(\theta)}\right) + \mathrm{erf}\left(\frac{\rho_{1,j,j+1}}{2\sigma_{rsj,j+1}\tan(\theta)}\right)\right]} \tag{8}$$

We consider shadowing effects only at the air-snow interface.

## 5 Texture Synthesis

The greatest flexibility in the study involved the generation of texture given average values of snow properties and backscatter coefficients at low-resolution over the terrain. In the absence of information about surface roughness characteristics (rms height, slope distributions, anisotropic correlation forms and lengths at multiple scales, etc.) we chose to use empirical studies of texture in various settings to guide our work. We have also attempted to devise a texture generation method general enough to accommodate the additional information provided by more sophisticated models as they become available.

"Texture" here is taken to mean the pattern of random spatial variation in intensity in the image. This may include the interferometric speckle inherent in coherent imaging systems, the variation in intensity due to incidence angle dependent scattering from a non-planar surface and the intensity variation due to variation in micro-scale physical properties in the terrain being imaged. Texture should then be characterized at multiple scales, with the smallest scale of interest being determined by the resolution of the imaging instrument – the pixel footprint. Representing the texture statistically, we confine our attention to the intensity distribution within the image, i.e. the image histogram, and the spatial correlation between pixel intensities within some neighborhood of a given pixel. The scale of the texture of interest determines the size of this neighborhood. The spatial variation of intensity is described by the *autocorrelation function* (ACF). With sufficiently sophisticated physical models, the autocorrelation function may follow from model predictions. In this study the texture shaping autocorrelation functions are input by the user.

## 5.1 Spatial Variation of Intensity - Correlation

In Phase I we have assumed that the user will specify the form of the autocorrelation function. The user may select from a range of ACF's, one form being the anisotropic 2-D generalized Gaussian:

$$R[l_1, l_2] = R_0 \exp\left\{ - \left( \left[ (l_1 \cos\theta + l_2 \sin\theta) / \alpha_1 \right]^2 + \left[ (-l_1 \sin\theta + l_2 \cos\theta) / \alpha_2 \right]^2 \right)^{\gamma} \right\} \quad (9)$$

Where $\alpha_1$, and $\alpha_2$ are the correlation lengths in two orthogonal directions, θ is the angle between the x-axis and the major axis of the ACF ellipse and γ is a parameter controlling the exponential vs. Gaussian character of the ACF. If γ is ½, the ACF is exponential, if γ is 1.0, the ACF is Gaussian. Values of γ between ½ and 1.0 are intermediate between exponential and Gaussian. Note that values of γ greater than 1.0 result in functions that are not valid ACF's (Yaglom (1987)). Another ACF form is the Gaussian-exponential sum:

$$R[l_1,l_2] = R_0 \exp\left\{-\left(\left[\left(l_1\cos\theta + l_2\sin\theta\right)/\alpha_1\right]^2 + \left[\left(-l_1\sin\theta + l_2\cos\theta\right)/\alpha_2\right]^2\right)^{1/2}\right\} + \quad .$$

$$(1.0 - R_0)\cdot\exp\left\{-\left(\left[\left(l_1\cos\theta' + l_2\ \sin\theta'\right)/\alpha'_1\right]^2 + \left[\left(-l_1\sin\theta' + l_2\cos\theta'\right)/\alpha'_2\right]^2\right)\right\} \qquad (10)$$

Here we allow the correlation lengths and preferred directions to differ for the Gaussian and exponential components of the ACF, with the Gaussian-exponential character controlled by the value of $R_0$. If $R_0$ is 1, the ACF is exponential. If $R_0$ is 0, the ACF is Gaussian

The ACF may vary over the full terrain, both in degree and direction of anisotropy, and, in the case of the generalized Gaussian (eqn. 9) and exponential-Gaussian sum (eqn. 10) forms, the degree to which the ACF is Gaussian vs. exponential. One part of the terrain may have anisotropic Gaussian ACF while another may have isotropic exponential ACF, with smooth variation in textures between the two points. See Fung, Appendix 2B (1994) for some discussion of surface correlations.

The use of 2-point statistics (autocorrelation functions) to simulate textures is supported by Oliver (1988) where they were applied to images with greater complexity than the snow-backgrounds we consider here. The support of anisotropic correlations allows the user to simulate textures affected by directional influences in the scene – wind formed structures or shadow effects on snow cover evolution, for example.

## 5.2 Image Intensity Histogram – The Intensity PDF

Given the particular correlation desired for the spatial variation of mmw image intensity, we must also decide on the image intensity histogram (an estimate of the intensity *p*robability *d*ensity *f*unction, *pdf*) we wish the simulated texture to display. The most recent extensive experimental work done by Ulaby et al. (1998) indicates that for 95 GHz. scattering at high angles of incidence, the Rayleigh fading model "provides an excellent fit to the measured data [backscattering cross-section per unit area, $\sigma_A$] for various types of terrain covers, including bare surfaces, grasses, trees, dry snow, and wet snow." The Rayleigh fading model assumes (1) that each pixel contains several scatterers, (2) these scatterers are randomly distributed within the pixel and (3) the strength of the returns from each of the scatterers are comparable in magnitude. This model requires statistically homogeneous scenes, which snow backgrounds provide. If we were to combine data from multiple types of terrain targets into a single image, the pdf of the combined data would no longer be exponentially distributed. In that case the K-distribution better describes the pdf of the resulting heterogeneous scene. The Rayleigh fading model for backscatter coefficients implies that $\sigma_A$ is exponentially distributed. Ruderman and Bialek (1994) have examined images of more complex backgrounds and found that the statistics of ensemble averages of images taken of woodland scenes exhibit nearly scale invariant contrast distributions with exponential tails, hence the images are not Gaussian. Although there is tremendous variability in intensity histograms from image to image, *ensembles* of these images have highly robust statistical features. It is this average that we will attempt to simulate, The result will be a textured scene that will accurately simulate an image consistent with the physical constraints provided by the user.

### 5.2.1 Backscatter Coefficient Random Field

We first assume that the user will specify the number of pixels used to depict texture on a given terrain. For example, if the full terrain is composed of 30 by 30 square tiles, the user may specify that each tile use $64^2$ pixels to represent the texture in that tile. Given the average backscatter coefficient, $\sigma_A$, (proportional to image intensity) for a particular terrain tile we must next assign coefficients, $\sigma_{i,j}$, to each of the $64^2$ pixels in that tile. We assign these coefficients to each of the pixels in the tile by interpolating (bilinearly) between the four nearest average backscatter coefficients, $\sigma_A$, computed with the EMSARS or geometric optics models.

If we follow Ulaby et al. (1998) and assume that the image intensity will be exponentially distributed, we must assign values to each of the pixels in the tile that will result in an exponential *intensity* distribution after correlation "shaping" has been carried out. Since our correlation shaping algorithm (see Section 5.1) operates on a 2-D complex field, that is, a random number $e_r + ie_i$, is associated with each pixel, we must select from an appropriate *Gaussian* distribution for $e_r$ and $e_i$ to produce the desired *exponential* distribution when computing the intensity, $e_r^2 + e_i^2$. We will use the coefficients, $\sigma_{i,j}$, to determine the distribution from which to draw the random field values $e_r$ and $e_i$ for each pixel.

For a given pixel, we want $E\lfloor e_r^2 + e_i^2 \rfloor = \sigma_{i,j}$. Let $e_r$ and $e_i$ be selected from the same zero mean Gaussian distribution. Then $E\lfloor e_r^2 + e_i^2 \rfloor = E\lfloor e_r^2 \rfloor + E\lfloor e_i^2 \rfloor = 2\upsilon$, where $\upsilon$ is the variance of the distribution. We therefore pick $e_r$ and $e_i$ from a zero mean

Gaussian distribution with variance υ = $\sigma_{i,j}/2$. In this way the bilinearly interpolated values of the backscatter coefficients at each pixel, $\sigma_{i,j}$, prescribe the random field over the full terrain. This procedure provides an uncorrelated random field which can then be shaped using the above discussed autocorrelation functions to produce the desired local intensity distribution (histogram). Figure 3b depicts a given terrain tile (the top of one snow column), with the *sntherm* and *scatter* predicted mean backscatter coefficients, $\sigma_A$, at points (1) through (4). The value of $\sigma_{i,j}$ at pixel *i,j* (point 5) is obtained by bilinearly interpolating the values of $\sigma_A$ at points (1) through (4). Particular values of $e_r$ and $e_i$ are then obtained for all points (1) through (5) by selecting from the random distributions illustrated in Figure 3a. The distributions of Figure 3a are actually symmetric about 0 – only one half of the distributions are depicted.

## 5.2.2 Correlation Shaping – Single Terrain "Tile"

Given a random field over the full terrain, we apply 2-D linear filters over the field, with correlation lengths and preferred direction varying from cell to cell over the terrain, to shape the correlations over the terrain. Assume for example, that the desired correlation function is of the form

$$R[l_1,l_2] = R_0 \exp\left\{\sqrt{[(l_1\cos\theta + l_2\sin\theta)/\alpha_1]^2 + [(-l_1\sin\theta + l_2\cos\theta)/\alpha_2]^2}\right\} \qquad (11)$$

where θ is the preferred correlation direction, $\alpha_1$ is the $e^{-1}$ correlation length in the preferred direction, $\alpha_2$ is the $e^{-1}$ correlation length orthogonal to the preferred direction and $R_0$ is the average power for the process. As per the procedure described in section 5.2.1, we generate a two-dimensional, complex Gaussian random field $f[n_1,n_2]$ with the following statistical moments:

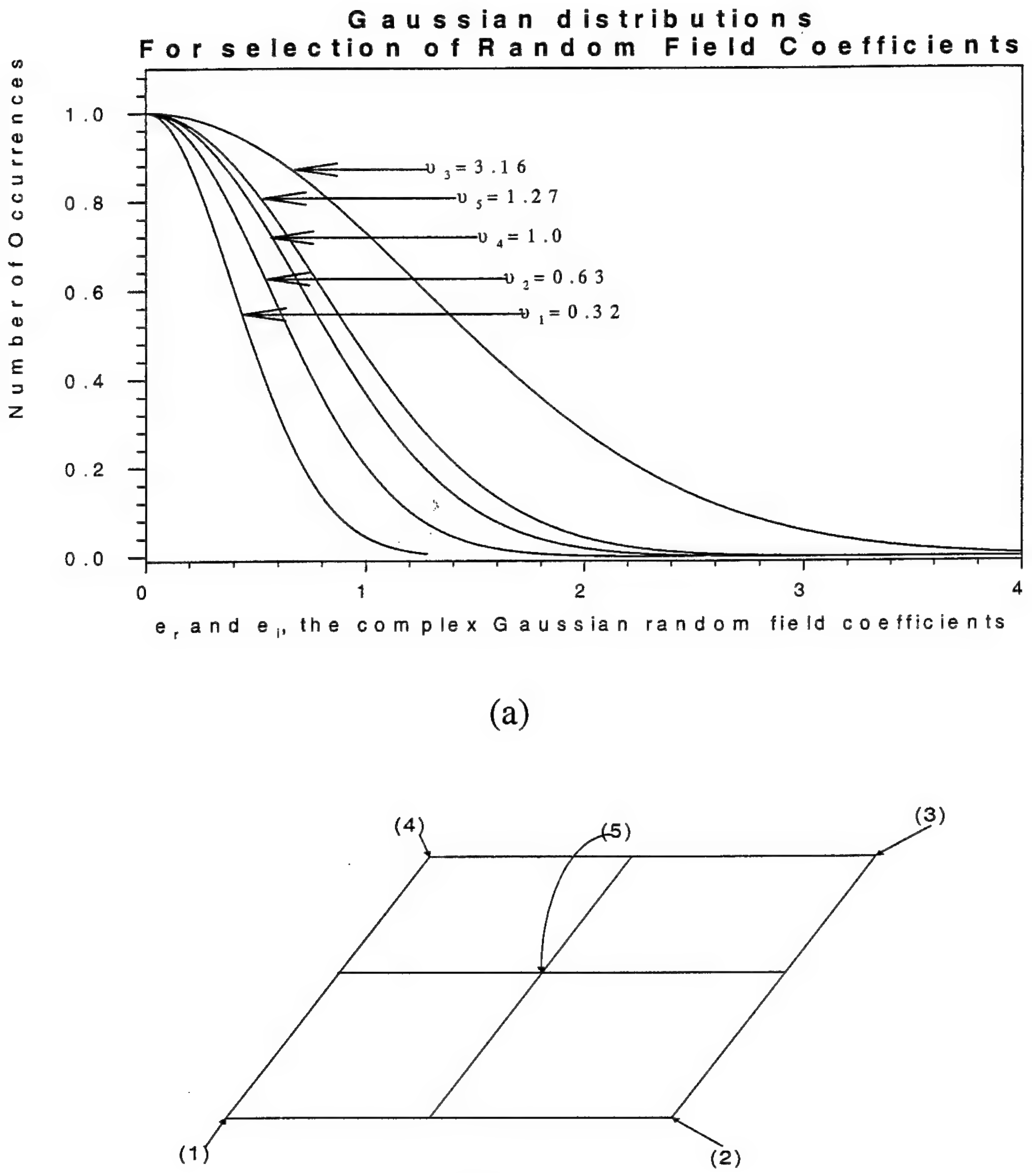
$$\mathrm{E}\left(f[n_1,n_2]\right)=0 \qquad (12)$$

$$R_f[l_1,l_2]=\mathrm{E}\left(f[n_1,n_2]f^*[n_1-l_1,n_2-l_2]\right)=\delta[l_1,l_2] \qquad (13)$$

and

$$\mathrm{E}\left(f[n_1,n_2]f[n_1-l_1,n_2-l_2]\right)=0 \qquad (14)$$

where $\delta[l_1,l_2]$is the two-dimensional Kronecker impulse function. The power spectrum for this process is the Fourier transform of its autocorrelation,



(a)

(b)

**Figure 3: (a) Distributions from which complex field values are drawn for points (1) through (5). The distribution variances ν are determined by interpolating between scattering coefficient values calculated for points (1) through (4). (b) A terrain tile with average scattering coefficient values computed at points (1) through (4) and interpolated at (5).**

$$S_f(\nu_1,\nu_2) = \sum_{l_1=-\infty}^{\infty} \sum_{l_2=-\infty}^{\infty} R_f[l_1,l_2] \cdot \exp[-j2\pi(l_1\nu_1 + l_2\nu_2)] = 1 \qquad (15)$$

so that the process is usually referred to as *white Gaussian noise.* If the process is filtered by a linear shift invariant system with frequency response $H(\nu_1,\nu_2)$, then the output process $g[n_1,n_2]$ will be a zero-mean, wide-sense stationary Gaussian process with the power spectrum,

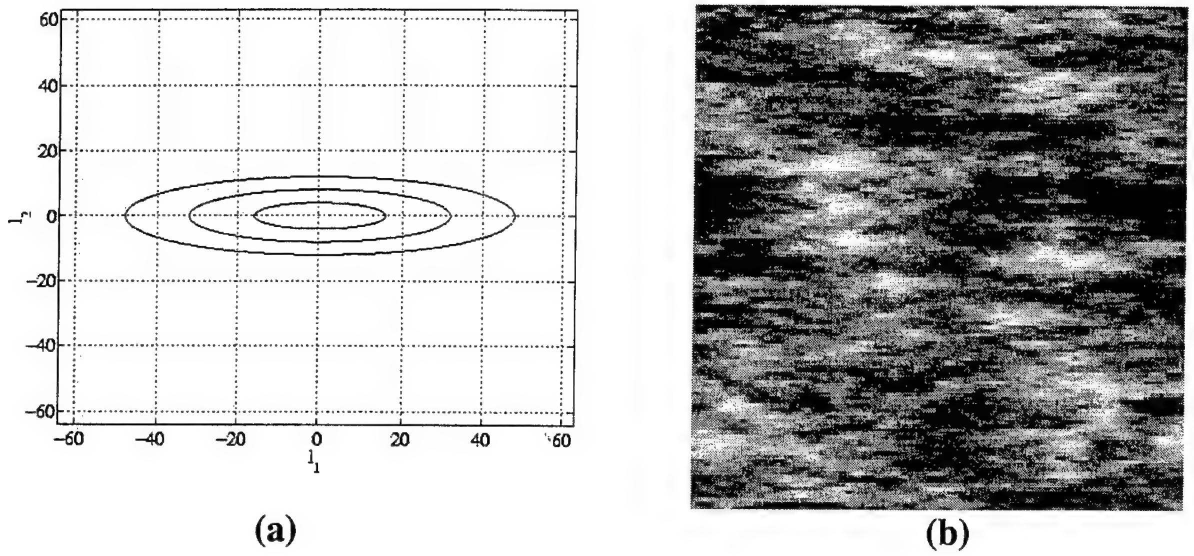
$$S_g(\nu_1,\nu_2) = |H(\nu_1,\nu_2)|^2 S_f(\nu_1,\nu_2) \quad .$$

$$= |H(\nu_1,\nu_2)|^2 \qquad (16)$$

If $H(\nu_1,\nu_2)$is selected to be the square-root of the Fourier transform of the autocorrelation function specified above,

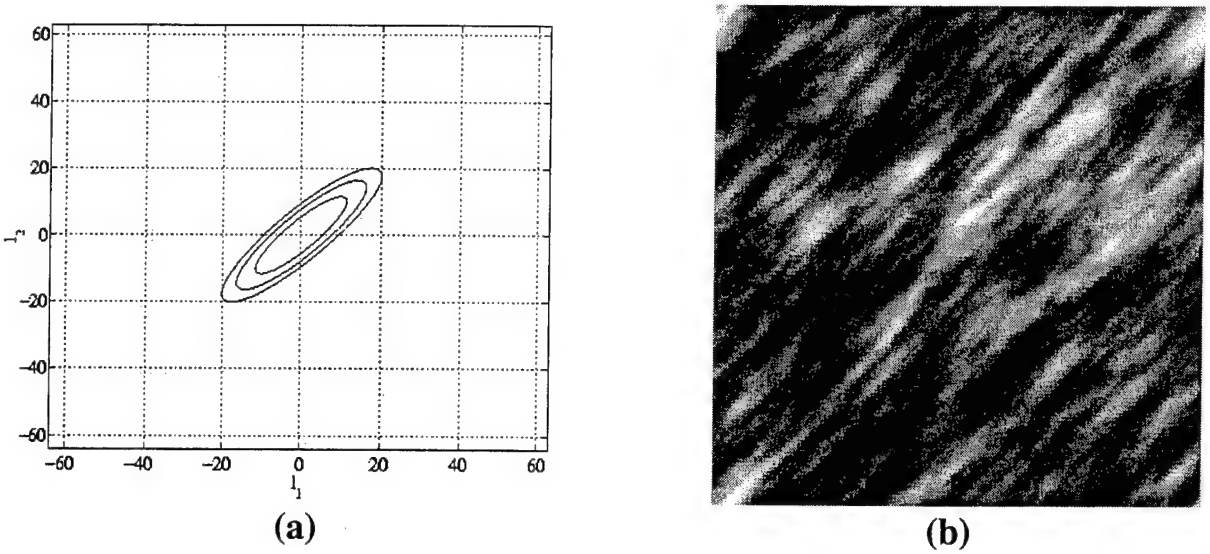
$$H(\nu_1,\nu_2) = \left( \sum_{l_1=-\infty}^{\infty} \sum_{l_2=-\infty}^{\infty} R[l_1,l_2]\exp[-\mathrm{j}2\pi(1_1\nu_1 + l_2\nu_2)] \right)^{\frac{1}{2}} \qquad (17)$$

the process $g[n_1,n_2]$ will have the desired correlation function. Figure 4 illustrates an example of anisotropic texture with exponential ACF. In this case the long correlation length is 16 pixels, the short length is 4 pixels and the major axis is directed along the horizontal axis.The contours of Figure 4a are at $e^{-1}, e^{-2}$ and$e^{-3}$. Figure 5 uses an ACF of form $R[l_1,l_2] = R_0\left[-\left\{[(l_1\cos\theta + l_2\cos\theta)/\alpha_1]^2 + [(-l_1\sin\theta + l_2\cos\theta)/\alpha_2]^2\right\}^{\gamma/2}\right]$ with $\alpha_1$=16, $\alpha_2$=4, γ = 1.5 and θ = 45 degrees. The ACF is intermediate between exponential and Gaussian. Again, the contours are at $e^{-1}, e^{-2}$ and$e^{-3}$. Finally, in Figure 6 the ACF is completely Gaussian, i.e. γ = 2. The correlation lengths are $\alpha_1$=16 and $\alpha_2$=4.



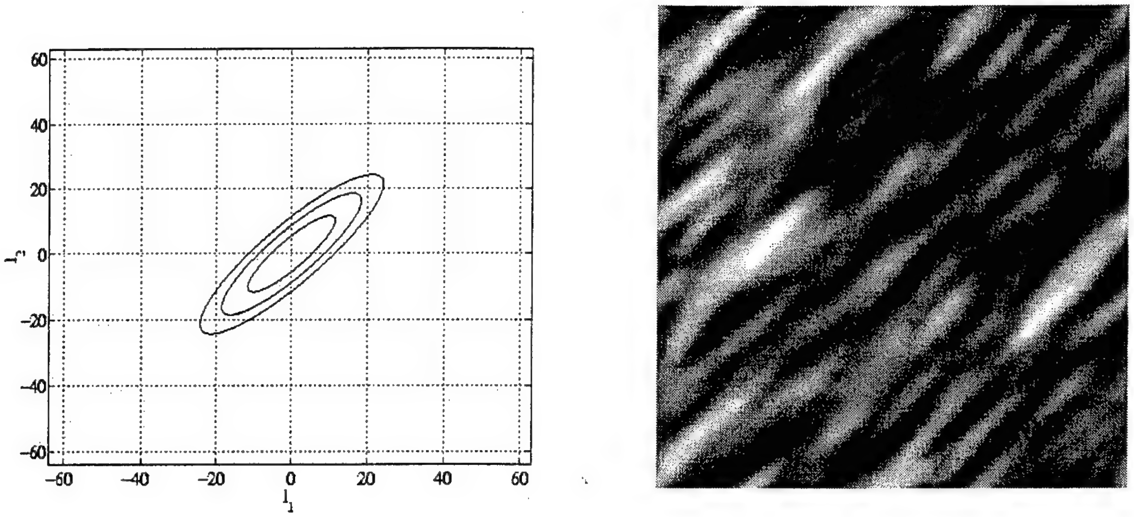
(a) (b)

**Figure 4: ACF Contour plots and example texture for exponential ACF with correlation lengths of 4 and 16 pixels and θ = 0. The texture was generated using a 128 by 128 field.**



(a) (b)

**Figure 5: ACF Contour plots and example texture for ACF with correlation lengths of 4 and 16 pixels and θ = π/4. The texture was generated using a 128 by 128 field**



(a)

(b)

**Figure 6: ACF Contour plots and example texture for Gaussian ACF with correlation lengths of 4 and 16 pixels and θ = π/4. The texture was generated using a 128 by 128 field.**

## 5.2.3 Correlation Shaping – Multiple Terrain "Tiles"

The procedure described above utilizes a single 2-D correlation function, $R[l_1, l_2]$, to shape the pixel intensity correlations over a given set of pixels. If we let the set of pixels be in the neighborhood of a given elevation data point ( i.e. the center of a terrain tile) we shape the correlations only in that neighborhood. In order to shape correlations over the full terrain, and allow the correlations to vary smoothly over the terrain, we must apply changing correlation filters to overlapping sets of pixels and then extract only a subset of the filtered pixels for display.

For example, let us divide the terrain into a 30 by 30 square grid and let each of the 30x30 "tiles" forming the full terrain surface be covered by 64x64 pixels. Also suppose that we want each of the tiles to exhibit different pixel intensity correlations, with the correlations to change smoothly from tile to tile. We may then select a 128 by 128 set of pixels (the complex Gaussian random field of section 5.2.1) in a square neighborhood centered in the tile and shape their spatial correlations. Roughly, we would do a 2-D Fourier transform on this set of pixels, multiply the transformed array by the Fourier domain correlation shaping filter and inverse Fourier transform the product. From this 128 by 128 set of pixels we take the center 64 by 64 pixels. This set of pixels will have the desired correlations and intensity pdf, and will be the set of "textured pixels" for the the terrain tile. We then repeat this procedure on the next tile with the particular correlation function desired for *that* tile.

We start with a $128^2$ set of pixels and extract the $64^2$ for final display so that pixels from adjacent tiles will influence the texture in the given tile. In this way discontinuities, "edges", in the texture will be smoothed from tile to tile. The dimensions of this oversized "filter window" (a 128 by 128 window for a 64 by 64 pixel tile in the above example) will depend on the correlation lengths, $\alpha_1$ and $\alpha_2$ in equation (11), of the particular correlation shaping function being used. Longer correlation lengths will require larger windows. If the window is too small for the correlation length, texture "edges" will be apparent at boundaries between tiles.

Figure 7 is an example textured "terrain". It is for a square terrain covered by 900 (30 by 30) tiles. The tiles are each covered with 32 by 32 pixels. The initial, unfiltered complex random field is generated by drawing from a unit variance, zero-mean Gaussian distribution. Each tile has a unique correlation. The correlations are constructed so that the upper left corner is a strictly isotropic, exponential correlation with $x$ and $y$ correlation lengths of 10 pixels. The texture gets increasingly anisotropic and Gaussian toward the lower right corner of the image – completely Gaussian with an ACF ellipse having an aspect ratio of 4 to 1 in the lower right corner. That is, strictly Gaussian with $x$ correlation length of 10 and $y$ correlation length of 40 pixels. The orientation of the ACF ellipse also varies throughout the image, as is evident in the image. A filter window of 128 by 128 was used and no texture 'edges' are visible at tile boundaries. Figure 7a was generated with a 128 by 128 filter window. Figure 7b used an 'undersized' window of size 32 by 32. The relatively small filter window with respect to the correlation length of the filter results in discontinuities at the tile edges.

This somewhat artificial example illustrates the flexibility available in the use of oversized linear "texture filters" to generate seamlessly varying textures across a terrain. More investigation is needed to allow a general specification of desired intensity pdf in the resulting texture. Johnson (1994) catalogues methods to generate one-dimensional correlated random sequences with specific distributions. These methods could be extended to the two-dimensional case of synthetic images. In Phase II we propose to implement a method to allow the specification of pixel intensity pdf from a wide family of distributions.

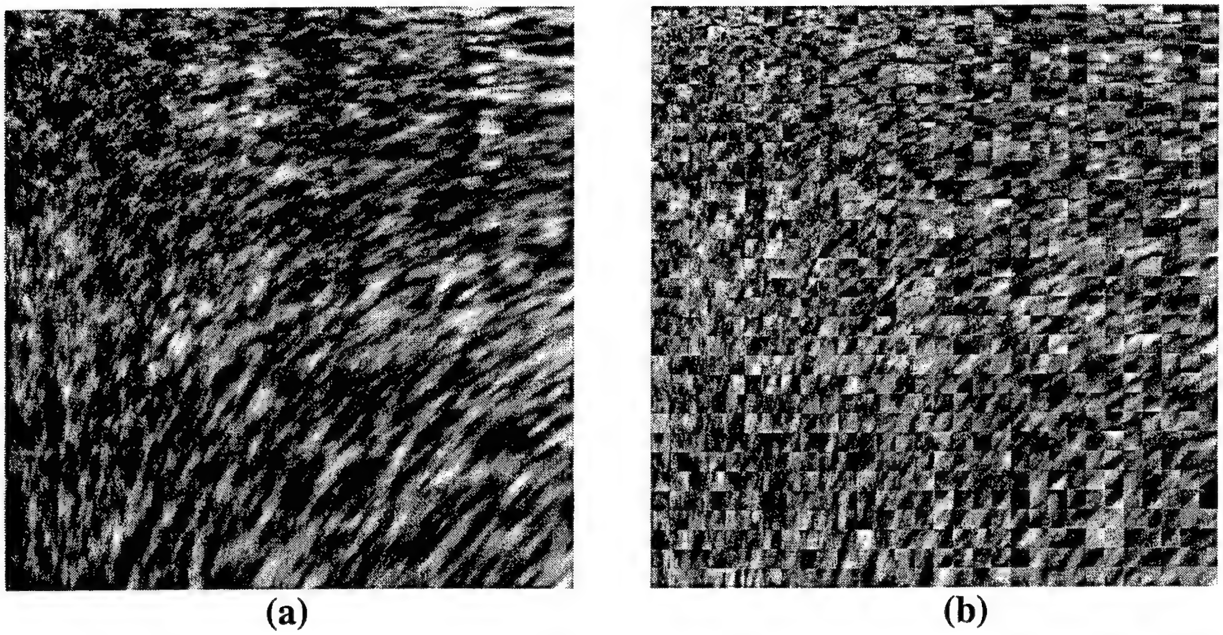


**Figure 7: (a) Example texture, 900 tiles, varying correlation functions. Note the absence of discontinuities at tile edges. (b) Use of undersized correlation shaping 'window' produces texture discontinuities at tile edges.**

### 5.2.4 Application of Texture to Terrain

With the random intensity variation determined over the full terrain, it remains to apply the resulting texture to the surface originally described by the elevation map. In Phase I we rely on the graphics features of OpenGL to treat the set of texture pixels as a "decal" that can be laid on a smooth surface generated by applying OpenGL "evaluator functions" to the original terrain elevation map. Evaluators make splines and surfaces that are based on a Bezier or Bernstein basis. After the surface is described in a spline basis, the set of intensity pixels derived through the above described procedure is then used to define a texture map that is applied to the spline described surface – using OpenGL texture mapping functions.

This procedure does not accurately retain the spatial statistics controlled with the autocorrelation functions, since application of the texture map to the spline-surface "stretches" the pixels over regions of finite curvature, changing the pixel sizes in these regions and thereby changing spatial correlations in those regions. Accurate application of the texture to 'displayable' surfaces, including the effects of surface curvature and perspective and range effects on pixel size, will be addressed in PhaseII.

## 6 Codes

We have included some of the computer code developed in Phase I. There are 4 programs in the suite. The first program, *surface*, uses terrain elevation and "class" data to generate a "tiled" surface. This surface is composed of triangles, two per input elevation point. *surface* also estimates surface unit normal vectors at each elevation point and writes a file with elevation, class and surface normal information to be used in following programs in the suite. It also generates a GTSIG compatible **.fac* file that can be modified for display with the FRED program.

Given terrain surface normal information (estimated with *surface), sntherm* initial conditions by terrain class (a "LAYER.IN" type file for each terrain class, see Jordan, 1990) and a meteorological data file, *sntherm* (a modified version of SNTHERM.89) calculates the evolution of snow with time for each snow column making up the full terrain. The results of the *sntherm* modeling are written to files for use in the scattering coefficient calculations.

The results of the terrain generation and snow modeling computations are used by program *scatter* to calculate the terrain surface scattering coefficients. These calculations require both the surface normal estimates generated by *surface*, and the snow property predictions generated with *sntherm*. The calculations also depend on the terrain surface normal variation and the observer's point of view vector.

After the average scattering coefficients have been calculated with *scatter*, we finally synthesize a texture. In addition to the average scattering coefficient, we need a set of parameters describing the desired spatial variation of the image intensity (the texture) for each terrain tile. We currently require the user to provide this information in a file. In the set of sample files included with this report, two sample files are provided – named *"corlenetc.dat"* and *"corlenetc1.dat'*.

## *6.1 Installation*

Included with this report is a 3½″ floppy disc with sample source code, executable code and input files. The code was written using Digital Visual FORTRAN Version 5.0 in a Windows 95 environment. The files were compressed using the WinZip utility.

To install the code, assuming the user is in a Windows 95 environment, place the diskette in the appropriate drive and use Windows Explorer to examine the contents of the diskette. The diskette holds a file named *install.exe*. To install the texture generation software, double-click on the *install.exe* icon (or choose Run from the Start menu and type A:INSTALL). The files are self-extracting, and the user will be asked for a directory in which to place the files. The default directory is *C:\TAI_Texture*. If the user wishes to place the files in another directory, he may enter the name of that directory. If he does so, however, he must modify the *shortcuts* in the work directory to reflect the location of the target files (the executables *surface.exe, sntherm.exe, scatter.exe* and *texgen.exe*) and the location of the work directory.

## 6.2 Common File

One file used as input to all four of the texture synthesis routines holds the names of all other input files. This is the file named "*filenames*". An example *filenames* file is shown here:

```
snow.in
metswe.in
undist2
snow.out
flux.
filt.out
terElev.dat
terElNorm.dat
ter.apa
observer.inp
SnowDenseRam.dat
FracVolWat.dat
tempRam.dat
thickRam.dat
grainRam.dat
grain.rad
FracVLiq.rad
FracVolI.rad
temp.rad
thickness.rad
scat.rad
 9.3999997E+10
corlenetc.dat
texture.pgm
```

The first line is the *base* name of the snow class information file. It corresponds to SNTHERM.89's *layer.in* file, but is combined with a 'snow column class number' to specify the snow information for all snow columns of a given class. For example, in the above *filenames* file, the base name is *snow.in*, so, if the snow at elevations 1, 21 and 67 were of class 2, the modified version of SNTHERM.89 would use the snow initial condition information in file *snow2.in* to model the snow evolution at those points in the terrain. The second line indicates the name of the meteorological data file used to drive *sntherm*. Lines 3 – 6 specify output file names for standard SNTHERM.89 output. They are not used by the texture synthesis codes but are left in to minimize the difference between the SNTHERM.89 *filename* file and the texture synthesis *filenames* file. Line 7 names the file holding the terrain elevation and snow class information. In Phase I the elevation data is assumed to be sampled at uniform intervals in both x and y directions. The first line of this file specifies the size of the elevation grid. Following this are the x (north), y (west) and z (elevation) coordinates and the 'snow column class number'. The units of the spatial coordinates are arbitrary in Phase I since the observer is assumed to be infinitely distant. That is, the observer's point-of-view vector is constant across the full terrain. Part of an example terrain elevation file is given below.

```
30   30
0.000000 0.000000 1.876670 1
0.000000 1.000000 1.919627 1
0.000000 2.000000 2.104806 1
0.000000 3.000000 2.266563 1
0.000000 4.000000 2.267572 2
0.000000 5.000000 2.316560 2
0.000000 6.000000 2.431220 2
              .
              .
              .
```

Line 8 of *filenames* is the name of the output file holding the surface normal vectors interpolated from data in the terrain elevation/class file. This file duplicated the information in the previously discussed file, but adds the surface normal information.

Line 9 of *filenames* specifies the name of the terrain apparent area file to be used to adjust direct solar insolation for shading of one part of the terrain by another. This file is not used, but the reader can examine subroutine *slope* in the *Sntherm* directory to see how it will be used in Phase II. The lines of code that utilize the apparent area information are commented out with the string *ctexture* in file *slope.f.* Line 10 is the name of the file holding the observer's point-of-view direction. The user specifies the azimuth in degrees ccw from north and zenith in degrees from "straight down". That is, if the observer were looking straight down on the terrain, the zenith angle would be 0 degrees. If the observer were looking at the terrain at a grazing angle, the zenith would be ~90 degrees. The following 4 file names specify output files for the modified SNTHERM.89 program. These files hold snow properties for each layer of each snow column. This information is used in the scattering coefficient calculations (see section 4.3). The next 5 files are *sntherm* output files to be used as input to the EMSARS based scattering coefficient calculations. These files can also be used with the MuSES software to display the snow properties of the terrain top layer. To display these properties with MuSES, the files need the extension *".rad"* and there must be a faceted terrain file for the terrain of interest. The next file name is the scattering coefficient calculation output file. In this example the file name has a *".rad"* extension – again to indicate that it is in an appropriate format for display with MuSES. Following this file name is the frequency, in Hz., of the radar being simulated. The last two file names are texture synthesis related. The first of these two files holds user supplied image texture parameters – x and y correlation lengths, the angle between the ACF oval principal axis and the terrain x-axis ($\theta$ in equation 9) and the ACF exponential/Gaussian parameter ($\gamma$ in equation 9). An excerpt from the supplied example file is given here:

```
Cor Length X    Cor Length Y     skew angle      gamma
   10.00000        10.00000       90.31000      1.000000
   10.00000        10.60118       90.00000      1.020039
   10.00000        11.27869       90.00000      1.042623
   10.00000        11.98153       90.00000      1.066051
   10.00000        12.69540       90.00000      1.089847
```

Finally, the synthetic texture output file name is given. This file should have the *".pgm"* extension, since the file is a *p*ortable *g*ray *m*ap format – displayable with Paint Shop Pro.

## 6.3 Running the Codes

After installing the texture generation files, the user may run example texture generation scenarios. The *C:\TAI_Texture\work* directory holds sample input files for texture synthesis. These files may be modified in order to see the effect of varying input on the resulting synthetic texture. To run the codes, use **Windows Explorer** to go to *C:\TAI_Texture\work* and simply double-click on the shortcuts. The first time through, the codes must be run in the order, (1) *surface*, (2) *sntherm*, (3) *scatter* and (4) *texgen.*

The user is reminded here that certain viewer positions, as specified in *observer.inp* (see the previous section), will result in parts of the terrain being obscured by other parts of the terrain. In Phase I, the resulting texture is not mapped to the original terrain, and is instead simply written to a 2-dimensional image file. Under many combinations of viewer direction and terrain surface normal, the angle of incidence will be greater than 90 degrees. This occurs when the viewer tries to look "through" one part of the terrain to see another part – looking through one side of a hill to see the other side, for example. In order to avoid the obscured portions of the terrain showing up as "blanks" in the 2-D texture image, an angle of incidence of 89.5 degrees with respect to the local terrain surface normal is simply assigned. In Phase II the mapping of the texture to the terrain elevation map will place the obscured parts of the texture image where the user won't be able to see them, and the assignment of an artificial angle of incidence will be unnecessary.

### 6.3.1 Running *surface*

To run *surface,* go to *C:\TAI_Texture\work* and simply double-click on the *surface* shortcut. *surface* will read *filenames*, input the file specified on line 7, the file holding terrain elevation and class information, and generate a file repeating this information and adding surface normal information (line 8 of *filenames*) and a GTSIG format facet file for display of the terrain surface named **.fac*.

### 6.3.2 Running *sntherm*

To run *sntherm,* go to *C:\TAI_Texture\work* and simply double-click on the *sntherm* shortcut. *sntherm* reads the meteorological data file (line 2 of *filenames*), the terrain normal vector file created by *surface*, and the initial snow condition files (line 1 of *filenames)* and produces the predicted snow condition files (lines 11 through 20 of *filenames*). The modified version of SNTHERM.89 included in this package also generates the standard SNTHERM.89 output file describing snow condition variation with time – line 4 of *filenames*. This file is written for each snow column making up the full terrain – overwriting the results for each previous column and thereby producing only one file for the full terrain. Some of the files generated – those named **.rad*, may be used to display the snow property variation with time, using the MuSES program and the **.fac* file generated by *surfa*ce.

### 6.3.3 Running *scatter*

To run *scatter,* go to *C:\TAI_Texture\work* and simply double-click on the *scatter* shortcut. We have provided code for the Narayanan/McIntosh, geometric optics based scattering calculations and not the EMSARS based code because of the extreme slowness and the greater complexity of the EMSARS code. We feel that additional work will be needed to integrate the EMSARS code into the texture synthesis package. *scatter* uses the snow property prediction files generated by *sntherm* (lines 11 – 15 of filenames) along with observer view direction information (line 10 of filenames) and a specification of the radar frequency, in Hz, (line 22 of filenames) to calculate radar backscatter coefficients for the layered snow columns predicted by *sntherm*.

Running *scatter* generates a file, named on line 21 of filenames, of radar backscatter coefficients,. The coefficients are written to a file in a format that MuSES may read and display, using the *.fac file generated with *surface*. These coefficients are used by *texgen* to generate synthetic texture.

### 6.3.4 Running *texgen*

To run *texgen,* go to *C:\TAI_Texture\work* and simply double-click on the *texgen* shortcut. *Scatter* computes the scattering coefficients and produces a file of coefficients for each terrain tile at each time at which *sntherm* has generated snow property predictions. These coefficients are used by *texgen* to synthesize an image texture file at the time step requested by the user. *texgen* will pre-read the scattering coefficient file, determine the number of times at which coefficients are available and report that number to the user. The user is then asked which set of coefficients (which time) to use to drive the texture synthesis.

As previously mentioned, *sntherm* and *scatter* do not provide surface statistic related information needed for texture shaping. For this reason the user must provide a file with this information. The image intensity histogram is controlled by *scatter* output, but the spatial variation of this intensity is controlled by four user specified parameters. These parameters are held in the file specified on line 23 of the *filenames* file. These parameters are: (1) the correlation length in the direction of the major axis of the 2-D autocorrelation function ellipse, (2) the correlation length orthogonal to the major axis, (3) the angle, in degrees, between the ACF major axis and the terrain x-axis (north) and (4) the parameter controlling the exponential vs. Gaussian character of the texture autocorrelation function. The correlation lengths are expressed in terms of the number of pixels at which the 2-D spatial autocorrelation function falls to $e^{-1}$. Anisotropic ACF's allow different correlation lengths in two orthogonal directions.

The Phase I version of the texture generation code does not use dynamic memory allocation and therefore requires that the integer parameters, *nxTiles* and *nyTiles* in the "include" file *tex.inc* be set equal to the number of terrain tiles in the *x* and *y* directions. In the case of the example files provided, these values are 30 x-tiles and 30 y-tiles. Changes in the number of tiles require changing the values in *tex.inc* and re-compiling. *texgen* compares the number of texture tiles specified in the *tex.inc* file, *nxTiles*nyTiles*, to the total number of scattering coefficients calculated by *scatter*. If these two values are different, *texgen* notifies the user.

The total number of pixels used to depict texture over the full terrain, in the x and y directions, is also specified in *tex.inc*, with parameters *X_PIXCNT* and *Y_PIXCNT*. The number of pixels used to depict texture in one tile, assuming equal numbers of pixels in the *x* and *y* directions for simplicity, is then approximately *X_PIXCNT/nxTiles.* Finally, the number of pixels in the correlation shaping window is specified with the parameter *nw*, also in *tex.inc*. At the very minimum, this parameter must be larger than the number of pixels in a given tile (i.e. *X_PIXCNT/nxTiles),* but it is also necessary to consider the correlation lengths when deciding on *nw*. As previously mentioned, if the window size is too small for a given correlation length, tile edges will be apparent in the synthetic texture. A rule of thumb is that the window size should be at least twice the largest correlation length to be encountered on the full terrain.

In Phase I the filter window must be square so that only one parameter is needed to define the window. Furthermore, the Fourier Transform implemented in *texgen* requires the window dimensions to be a "power of 2" - 16 by 16, 32 by 32 etc. In

Phase II we will implement a mixed-radix, 2-dimensional complex FFT that will allow windows of any dimension without the use of zero-padding.

Although we have assumed an exponential image intensity pdf, we have allowed for other intensity distributions as well. Ulaby and Dobson (1989) provide backscatter variability data for mmw scattering coefficients for wet and dry snow. This data is expressed as standard deviation of the coefficients assuming a log-normal distribution. The standard deviation is a function of the angle of incidence of the signal. In Phase II we will provide an option to the user to interpolate into this backscatter coefficient vs. angle of incidence data and specify an approximate log-normal distribution for the (speckle-free) intensity.

## 6.4 Modification of Sample Source Code

If the user has Digital Visual FORTRAN Version 5.0, he may modify the sample source code provided. To do so, use Windows Explorer to examine the directory that the source code is in – *C:\TAI_Texture\Scatter*, for example – and double click on the icon associated with the **.dsw* file. Microsoft Visual Studio will open up the workspace and the associated **.f* files and allow the user to modify and recompile the program.

Some modifications may require a change in project settings. For example, if the user increases the number of pixels used to represent texture in a single terrain tile, the user may need to increase the stack size – using the *output* category under the *Link* tab.

# 7 REFERENCES

1. Bohren, Craig and Donald Huffman, *Absorption and Scattering of Light by Small Particles,*Wiley-Interscience, New York, 1998
2. Chandrasekhar, S. *Radiative Transfer*, Dover, New York, 1960
3. Cole, K.S. and R.H. Cole, Dispersion and Absorption in Dielectrics Part I: Alternating Current Characteristics, *J. Chem. Phys.,* Vol. 9, pp. 341-351, 1941
4. Curran, A.R., et. al. Users Manual for PRISM 3.2, Keweenaw Research Center, Michigan Technological University, Houghton, MI, June 1996.
5. Fung, Adrian, *Microwave Scattering and Emission Models and Their Applications*, Artech House, Boston, 1994
6. Goodman, Joseph, *Statistical Optics,* John Wiley and Sons, New York, 1985
7. Jakeman, E. and P. Pusey, Significance of K Distributions in Scattering Experiments, *Physical Review Letters,* Vol. 40, pp. 546 – 549, 1978
8. Jakeman, E. and Tough, R., Non-Gaussian Models for the Statistics of Scattered Waves, *Advances in Physics*, Vol. 37, No. 5, pp. 471 – 529, 1988
9. Johnson, G.E., Constructions of Particular Random Processes, *Proceedings of the IEEE.* Vol. 82, pp. 270 – 285, 1994
10. Kostinski, A.B. and A.R. Jameson, Fluctuation Properties of Precipitation. Part 1: Deviations of Single Size Drop Counts from the Poisson Distribution, *Journal of the Atmospheric Sciences,* pp. 2174 - 2186, 1997
11. Narayanan, Ram and Robert McIntosh, Millimeter-Wave Backscatter Characteristics of Multilayered Snow Surfaces, *IEEE Trans. Antenna Propagat.,* Vol. 38, No.5, May 1990
12. Nghiem, S.V., M. Borgeaud, J.A. Kong and R.T. Shin, *Polarimetric Remote Sensing of Geophysical Media with Layer Random Medium Mode,* in Progress in Electromagnetics Research (PIER 3*),* Elsevier, New York, 1990
13. Oliver, C.J., the Representation of Correlated Clutter Textures in Coherent images, *Inverse Problems,* Vol. 4, pp. 843-866, 1988
14. Patankar, S. V., *Numerical Heat Transfer and Fluid Flow*, Hemisphere Publishing, New York, 1980
15. Ruderman, Daniel and William Bialek, Statistics of Natural Images: Scaling in the Woods, *Physical Review Letters,* Vol. 73, No. 6, August 8, 1994
16. Tsang, Leung, Jin Kong and Robert Shin, *Theory of Microwave Remote Sensing,* John Wiley and Sons, New York, 1985
17. Ulaby, Fawwaz, Adib Nashashibi, Alaa El-Rouby, Eric Li, Roger De Roo, Kamal Sarabandi, Ronald Wellman and H. Bruce Wallace, 95-Ghz Scattering by Terrain at Near-Grazing Incidence, *IEEE Trans. Antenna Propagat.*, Vol 46, No 1, Jan. *1998*
18. Ulaby, Fawwaz and M. Dobson, *Handbook of Radar Scattering Statistics for Terrain,* Artech House, Boston, 1989
19. Waterman, P.C. and R. Truell, Multiple Scattering of Waves, *Journal of Mathematical Physics,* Vol. 2, pp. 512-537, 1961
20. Yaglom, A.M. , *Correlation Theory of Stationary and Related random Functions,* Springer-Verlag, 1987